# Properties and Constructors

The objective of this exercise is to consolidate your understanding of C# properties and constructors.

|  |  |
| --- | --- |
| 1 | Create a new Class Library project called **CarLibrary** in the {installedFolder}\Labs\08\_Properties\_and\_Constructors\Begin folder.  Rename **Class1.cs**to **Car.cs** |
| 2 | Add a Console Application project to the Solution called **CarConsole**.  Set **CarConsole** as the start-up project.  In **CarConsole**, add a project reference to the **CarLibrary** project. |
| 3 | In **Car.cs**, create a property of type *int* called **Speed** with a *backing field* **speed.**  Validate that the speed set is above zero but under 100. |
| 4 | Add an auto-implemented property of type *string* called **RegistrationNumber**. |
| 5 | Add a calculated expression bodied property called **SpeedInKilometres**of type *double*.  To calculate the speed in kilometres, multiply the speed by 1.609344 |
| 6 | Add string properties for **Make**, **Model**, and **Colour**. |
| 7 | In **CarConsole**, in **Program.cs**:  Delete the line of code that outputs ‘Hello, World!’  Instantiate a car object, **c1**.  Issue a using directive to bring the **CarLibrary** namespace into scope.  Write the name of the instance to the console:  Console.WriteLine(nameof(c1));  Build and run the console applicationto confirm the object can be successfully instantiated.  Set the make of **c1** to be ‘**Ford**’.  Write the *make* of c1 to the console.  Write the *model* of c1 to the console.  What value is displayed? |
| 8 | In the **Car** class, create a constructor that accepts a *make* and a *model* only.  Initialise these values within the constructor. |
| 9 | In **CarConsole**:  Re-run the app. Does it build successfully? |
| 10 | Create a *parameterless* constructor  Set the make and model to be **Unknown**and the colour to be **Black**.  Confirm the console app builds and runs successfully.  What value is displayed for the model? |
| 11 | In **CarConsole**:  Instantiate a car object, **c2**, using the overloaded constructor. The make is **Audi**, the model is **TT**;  Write the make and model of **c2** to the console.  Set the colour property to **Red**.  Write c2’s colour property to the console.  Set the speed of **c2** to **30 miles per hour**.  Display the speed in the console in both *miles per hour* and *kilometres per hour*. |
| 12 | In **CarConsole**:  Instantiate a car object, **c3**, using the overloaded constructor (**BMW**, **X5**) and an object initialiser that sets the colour to **Grey** and the registration number to **ABC 123**.  Write the property values of **c3** to the console. |
| 13 | In **Car.cs**, chain the parameterless constructor to the overloaded constructor, passing **Unknown Make** and **Unknown Model** as the parameters.  In the body of the parameterless constructor, remove the make and model and set the colour to be **White**.  Confirm the console application still builds and runs successfully. |
| 14 | In **CarConsole**:  Instantiate a car object, **c4**, using the parameterless constructor.  Write the property values of **c4** to the console.  Run the project and confirm **c4** is an unknown make and model that is white, with an empty registration number. |